

17/03/2018

Thomas Legris & Guilpain Léo

ESIR2 - IOT

RSF – Les réseaux Wifi

Compte rendu

*« J’atteste que ce travail est original, qu’il indique de façon appropriée tous les emprunts, et qu’il fait référence de façon appropriée à chaque source utilisée »*

Table des matières

[Introduction 2](#_Toc509756065)

[Travail préparatoire 2](#_Toc509756066)

[Question n°1 2](#_Toc509756067)

[Question n°2 2](#_Toc509756068)

[Analyse de trafic 3](#_Toc509756069)

[Question n°1 3](#_Toc509756070)

[Question n°2 3](#_Toc509756071)

[Question n°3 4](#_Toc509756072)

[Question n°4 4](#_Toc509756073)

[Question n°5 5](#_Toc509756074)

[Allocation dynamique de canaux 5](#_Toc509756075)

[Question n°1 5](#_Toc509756076)

# Introduction

Le but de ce TP est d’étudier les réseaux locaux sans fils à travers la couche physique et liaison. Nous utiliserons pour cela plusieurs outils et programmes pour l’analyse des périphériques. Cela va nous permettre d’identifier les points d’accès les plus efficaces ou bien de regarder les différentes trames spécifiques de la couche liaison de données.

# Travail préparatoire

Tout d’abord, nous avons utilisé le logiciel Wireshark sur la machine Linux afin d’analyser les différentes trames qui passaient sur le réseau.

## Question n°1

Pour identifier le périphérique réseau, il suffit de taper la commande « ifconfig » afin d’obtenir tous les composants réseau, on peut y trouver l’interface wifi wlan0.

On obtient le nom de notre carte :



Figure 1 : Nom de la carte

## Question n°2

La commande « iwconfig » nous permet d’afficher tous les réseaux Wifi présent sur l’ordinateur. Elle nous permet également de faire passer la carte sans fil en mode monitor, on peut ainsi lire ce qui se passe dans le réseau.

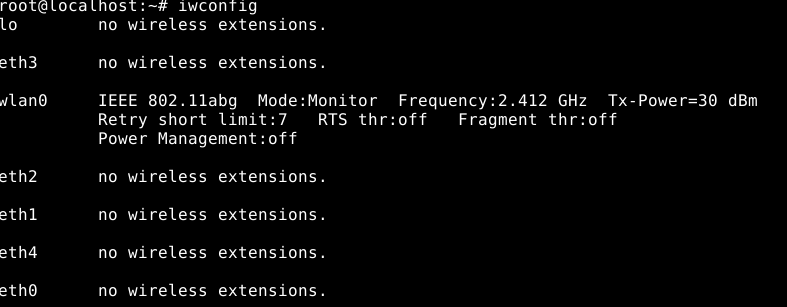


Figure 2 : Commande « iwconfig »

Sur cette figure, nous pouvons voir que seul l’interface wlan0 est sans fil. Il est configuré à la fréquence 2.4 Ghz. Pour obtenir la liste des canaux que l’on peut utiliser, nous devons entrer la commande « iwlist » suivi du nom de notre interface.

# Analyse de trafic

## Question n°1

Les trames *beacons* sont générées par les différents points d’accès en mode broadcast. Ces trames sont envoyées périodiquement permettant d’indiquer aux stations qu’un points d’accès sans fils est présent.

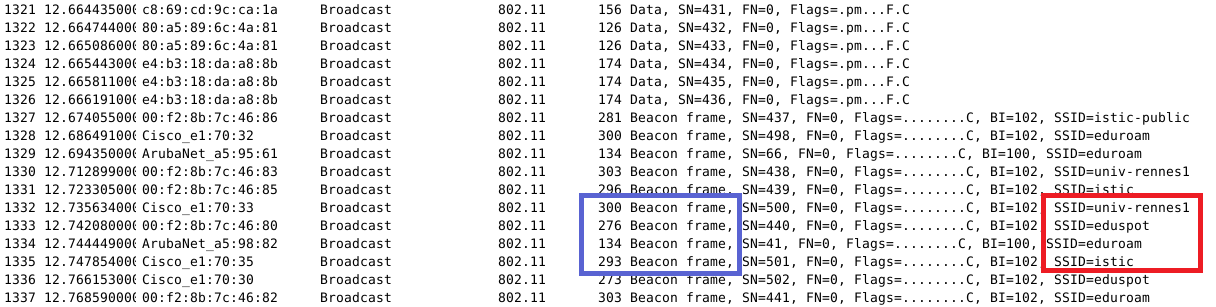


Figure 3 : Capture Wireshark des trames beacon

Sur cette capture wireshark, on observe que les différents points d’accès de l’université envoient régulièrement des trames *beacons* en broadcast. Elles sont reçues par notre station.

## Question n°2

L’étude d’une trame nous montre les couches suivantes :

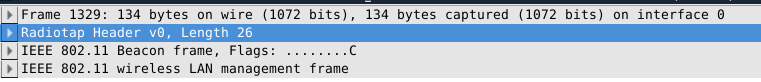


Figure 4 : Différentes couches d'une trame

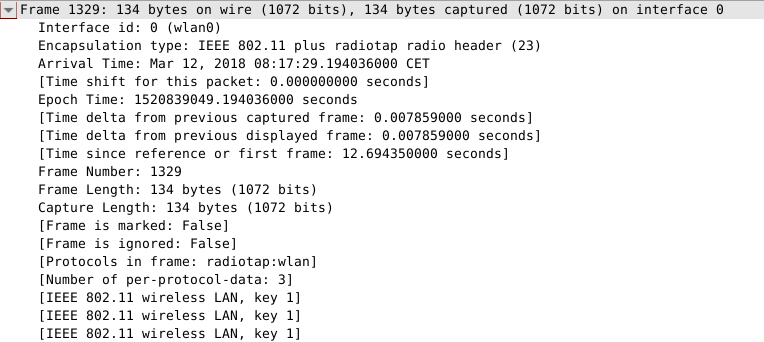


Figure 5 : Frame 1329

Le champ *Frame 1329* nous donne toutes les informations sur la trame. Cette couche contient notamment la date d’arrivée ainsi que la longueur de la trame.

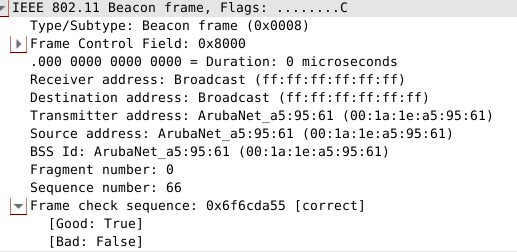


Figure 6 : Couche 802.11

On trouve aussi la couche *802.11 Beacon frame* qui contient d’autres informations propres aux fonctions du Beacon. En effet, cette trame est transmise en broadcast par un point d’accès qui est identifié comme étant la source.

On voit apparaitre :

* **Receiver Address** : l’adresse de la station à laquelle cette trame est envoyée (utile lorsque la trame doit transiter par des relais avant d’atteindre sa destination) ;
* **Transmitter Address** : l’adresse de la station expédiant la présente trame (utile lorsque cette station est une station relais).
* **Destination Address :** l’adresse du destinataire des données contenues dans le corps du paquet transmis ;
* **Source Address** : l’adresse de la source des données contenues dans le corps du paquet transmis ;

## Question n°3

Dans les nouvelles mises à jour des cartes, il n’y a pas de mode de gestion d’énergie, nous ne pouvons pas identifier les éléments correspondants. On peut voir sur le screen ci-dessous que le mode gestion d’énergie n’est pas supporté.

Une image contenant ciel

Description générée avec un niveau de confiance élevé

Figure 7 : Mode économie d'énergie

## Question n°4

Figure 8 : Visualisation des différentes trames

Il y a différentes trames qui sont analysées. Par exemple, il y a des trames beacon, des trames aknowledgment, des trames Data.

## Question n°5

Lors de la connexion d’une station à un points d’accès, il y a plusieurs étapes permettant l’authentification.

Premièrement, il s’agit d’identifier notre téléphone. Nous avons utilisé le smartphone possédant l’adresse MAC Wifi 94 :65 :2d :7c :89 :71.



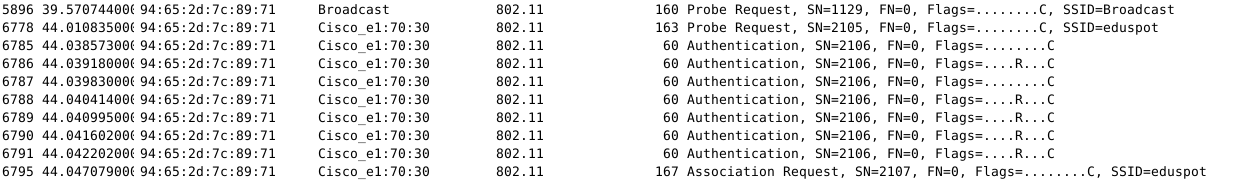


Figure 9 : Résultat de l'authentification

La première étape est « Probe request », elle permet au téléphone de rechercher les différents réseaux aux alentours. Une fois le réseau trouvé (ici tentative de connexion au réseau universitaire « eduspot ») l’appareil fait une demande d’authentification, une fois cette phase terminée, le téléphone cherche alors à s’associer avec le point d’accès.

# Allocation dynamique de canaux

## Question n°1

Cette partie du TP vise à gérer les RF dynamiquement. Tout d’abord, nous avons créé un programme Python afin de permettre au client d’obtenir le meilleur point d’accès.

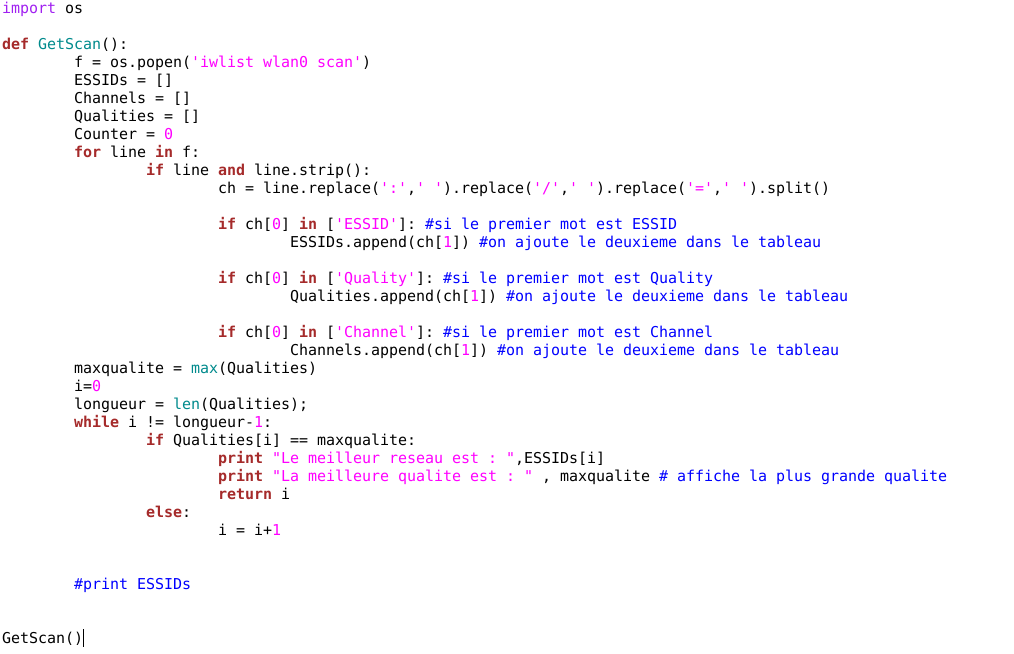


Figure 10 : Programme permettant d'obtenir le nom du réseau

Lorsque l’on fait un scan sur wlan0, on obtient de nombreux éléments concernant les différents réseaux.

Une image contenant texte

Description générée avec un niveau de confiance très élevé

Figure 11 : iwlist scan wlan0

Comme on peut le voir, on a accès au canal, à la fréquence, à la qualité et au nom du réseau. Pour choisir le réseau le plus adéquat, il faut trouver celui dont la qualité est maximale. Le programme ci-dessus permet de renvoyer le nom du réseau qui a la qualité maximale ainsi que sa qualité.

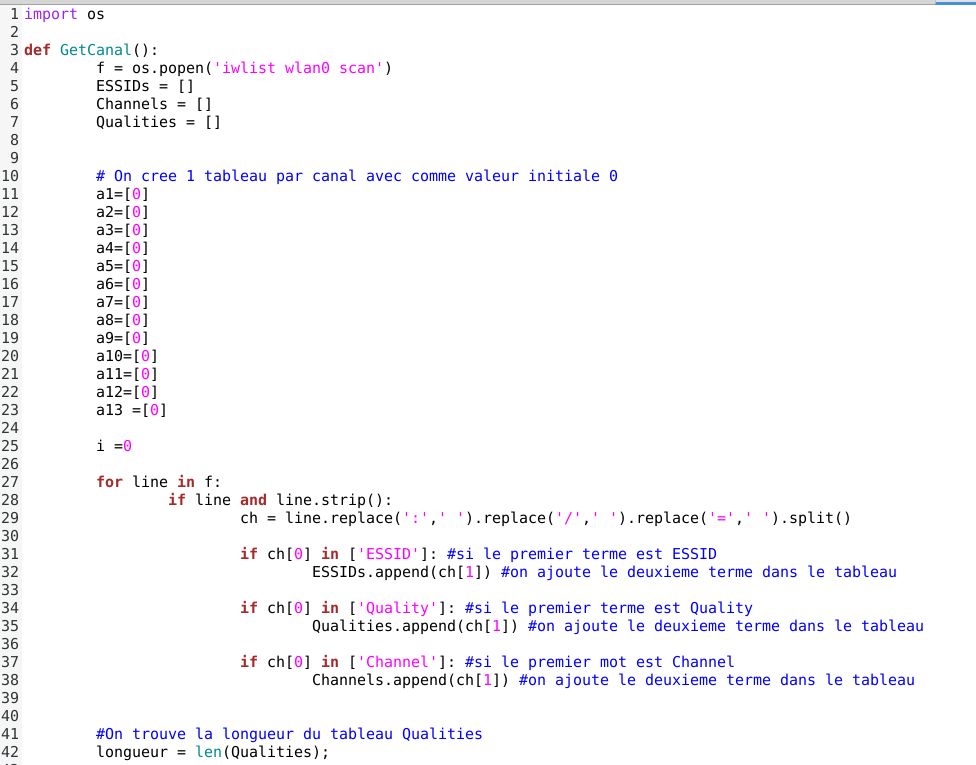
Tout d’abord, on définit le fonction « getScan() » en initialisant 3 listes : *ESSIDs*, *Channels* et *Qualities*

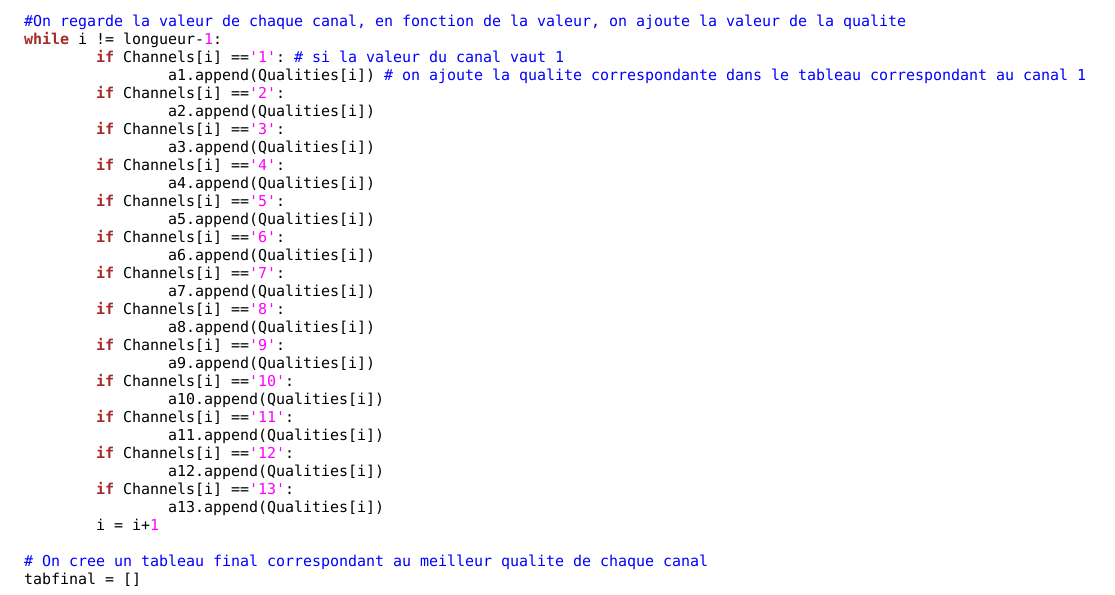
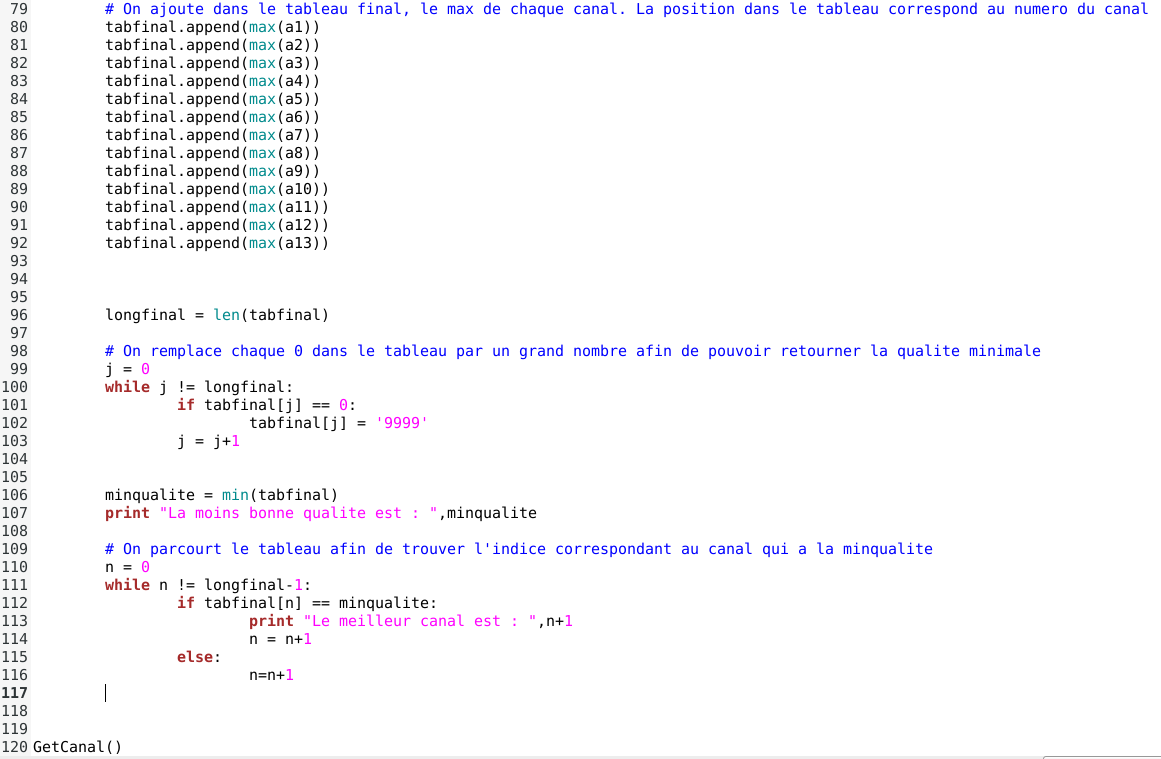
Pour parcourir chaque réseau wifi public, on scan l’interface wlan0 et on parcourt celui-ci afin de lister chaque nom de réseau et chaque qualité. Déterminer l’élément le plus grand dans une liste est très facile, il suffit d’utiliser la fonction max() sur cette liste, qui nous retourne la meilleure qualité. On refait alors une boucle afin d’associer le réseau qui correspond à cette qualité.

Voici le résultat de notre code.

Figure : Résultat du code

## Question n°2





Voici le code pour notre programme. Ce dernier permet de retourner le meilleur canal en fonction de la qualité.

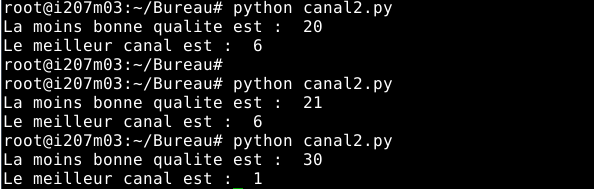


Figure : Résultat du code

Nous avons également tenté de traiter les entrelacements. Nous avons réalisé ce code mais nous n’avons pas pu le tester.

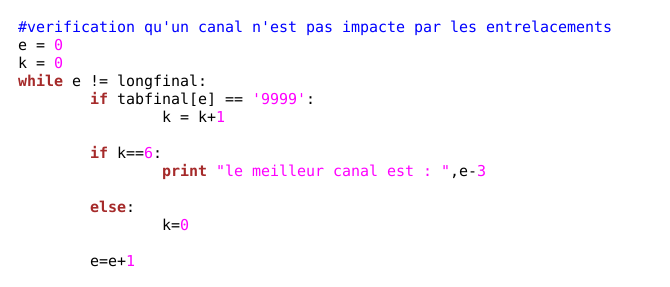


Figure : Test entrelacement